



Probando lista #1

Transcripción

[00:00] Bien, estamos aquí con los parámetros que hemos retornado de nuestro registros de la entidad médico que están en la base de datos, pero nuestro problema ahora es que estamos retornando más datos de los que deberíamos. Vamos paso a paso. ¿Qué tenemos aquí? Lo primero es que la URL de aquí, el request que hemos creado aquí es diferente el request que teníamos en post.

[00:27] Porque primero el verbo es diferente. Este es un get, este es un post. Post es para salvar datos, el get es para traerlos. Segundo, vemos que no estamos enviando ningún tipo de body porque como estamos pidiendo datos, no tenemos que enviarle algún tipo de dato que vaya a persistir a nivel de base de datos.

[00:49] Tercero, y no menos importante, vemos que la lista ya llega formateada en JSON, en formato JSON. Spring automáticamente entiende que por defecto necesitamos un JSON pero en configuraciones más avanzadas esto puede ser incluso formateado a XML, a texto, esos ya son pasos un poco más avanzados en Spring, por ahora estamos en lo básico.

[01:11] Tenemos una lista que está llegando con todos los parámetros tal cual está en nuestro modelo de base de datos en nuestra entidad, y esto no es bueno. ¿Por qué? No es bueno porque estamos exponiendo más información de la que deberíamos y quizás por regla del negocio, por motivos de seguridad, algún tipo de información privilegiada por ejemplo, no deberíamos ser capaces de mostrar ciertos datos.

[01:38] Por algo el negocio se toma el tiempo de definir las reglas y decirnos qué debería ser retornado. ¿Qué es lo que hacemos ahora? Bueno, de la misma forma en la que usamos un `datosRegistroMedico` para representar la información que llega a nuestra url, que no necesariamente es la misma que está en nuestro modelo de base de datos, para retornar esa información y para mostrarla podemos usar un DTO también.

[02:08] Entonces vamos a crearle aquí un DTO. En este caso va a ser si aquí tengo `datosRegistroMedico` aquí voy a poner `datosListadoMedicos`. Ese DTO no existe entonces el control de espacio, vamos a darle aquí y le vamos a poner `create record datosListadoMedico`. Controller es directamente en médico. Punto médico, correcto.

[02:34] Voy a ir. Creamos nuestro nuevo record, ya sabemos que internamente Java a crear una clase con esto, lo que tenemos que hacer aquí es especificar los parámetros que queremos que se muestren. Vamos de vuelta a nuestras reglas del negocio y necesitamos nombre, especialidad. Entonces vamos aquí, `String nombre`, `String especialidad`, `String documento`, `String email`. Todo separado por comas.

[03:17] Y listo, venimos aquí y ahora vemos que `datosListadoMedico` ya está aquí presente, pero ahora no compila porque este `medicoRepository.findAll` retorna un listado de médico como entidad, no de `datosListadoMedico`. Para esto yo voy a hacer uso del API Stream de Java, entonces aquí le voy a dar `stream.map` y le voy a decir que use los datos de `listadoMedicos` y que me cree un nuevo `datosListadoMedico` con cada médico que me traiga de la base de datos.

[04:00] Y a esto lo voy a poner un `toList` para que retorne en un objeto del tipo de lista. Vemos que no está compilando porque falta un constructor aquí, que use el parámetro del médico entidad para mapearlo a médico DTO. Vamos a crearlo, no hay ningún problema con eso. Venimos aquí y le vamos a dar un constructor. Entonces comenzamos aquí con el constructor.

